

Relaxation of Distributed Data Aggregation for Underwater Acoustic Sensor Networks

M. Rabbat
M. Coates
McGill University

Prepared By:
McGill University
845 Rue Sherbrooke O
Montréal, QC H3A 0G4 Canada

Contractor's Document Number: AMBUSH.1.2
Contract Project Manager: M. Rabbat, 514-398-4455
PWGSC Contract Number: W7707-145675
CSA: Stephane Blouin, DRDC - Atlantic Research Centre, 902-426-3100 x216

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of the Department of National Defence of Canada.

Contract Report
DRDC-RDDC-2015-C022
March 2014

- © Her Majesty the Queen in Right of Canada, as represented by the Minister of National Defence, 2014
- © Sa Majesté la Reine (en droit du Canada), telle que représentée par le ministre de la Défense nationale, 2014

Relaxation of Distributed Data Aggregation for Underwater Acoustic Sensor Networks

Contract Report # AMBUSH.1.2

Contract # W7707-145675

M. Rabbat, M. Coates

McGill University (Montreal, QC, Canada)

Fiscal Year 2013-2014

March 31 2014

This page intentionally left blank.

Table of contents

| | |
|---|----|
| Table of contents | i |
| List of figures | ii |
| List of tables | ii |
| 1 Introduction | 1 |
| 2 Problem formulation | 1 |
| 3 Methodology | 2 |
| 3.1 Gossip algorithms for distributed averaging | 3 |
| 3.2 Distributed particle filtering | 4 |
| 3.3 Distributed ensemble Kalman filtering | 5 |
| 4 Results | 6 |
| 4.1 Distributed particle filtering | 6 |
| 4.2 Distributed ensemble Kalman filtering | 9 |
| 5 Summary and ongoing work | 10 |
| References | 12 |

List of figures

- Figure 1: (a) Trajectories of the source and the sensors in the experiment considered. (b) Time-averaged squared error vs. reception probability p for a particle filter using both bearings and range measurements. (c) Time-averaged disagreement vs. reception probability p . The results shown correspond to 1000 Monte Carlo trials. 7
- Figure 2: Performance of the centralized particle filter with the number of particles ranging from 10 to 1600. (a) Root mean-squared error vs. number of particles. (b) Runtime vs. number of particles. (c) Example tracking performance under the two different motion models for the centralized particle filter with 1600 particles. 9
- Figure 3: RMSE of the particle filter under the CV+CT model. Red segments indicate times where the RMSE peaks. These coincide with times when the noise source turns towards the sensors and its trajectory is roughly co-linear with the sensors. 10

List of tables

- Table 1: Tracking Performance for the Distributed Ensemble Kalman Filter . . . 11

1 Introduction

This aspect of the project is concerned with coordinating the sensors in an underwater acoustic network in order to collaboratively track an acoustic source. Measurements are taken at each sensor node, and in order to obtain the best accuracy, the measurements should be jointly processed or fused. This requires communication and coordination among the nodes. At the same time, underwater communication is notoriously challenging. Channel conditions change rapidly and high data-rate communications are generally not possible. Consequently, protocols and mechanisms must be used which can adapt to the time-varying and unreliable communication medium.

Tracking strategies that rely on relaying all measurements to a central location become infeasible, both because of communication overhead and robustness concerns. We therefore focus on decentralized methods in which the microprocessor system attached to each sensor performs tracking. These local tracking algorithms share information between neighbouring sensors, distilling the raw measurements into essential summary statistics so that the communication burden is reduced.

In this first year of the project, we have focused on developing distributed algorithms for single-target tracking. In the second year, we will migrate our attention to the multi-target scenario. We have strived to keep our research activities tied to realistic measurement environments by repeatedly assessing performance using experimental datasets. The main application of interest involves bearings-only measurements, but we have also considered other measurement scenarios in order to make a more generalizable research contribution. We have explored and developed two algorithmic methodologies, concentrating on tracking solutions that can adequately address the non-linearities both in target dynamics and sensor measurements.

2 Problem formulation

Our goal is to track the state of a single noise source using measurements gathered from a distributed network of sensors. We denote the state of the noise source at time t by \mathbf{x}_t . Typically the state is a four-dimensional vector, with two dimensions for the source position and two dimensions for the source velocity. The state evolves according to a dynamic model,

$$\mathbf{x}_{t+1} = g(\mathbf{x}_t) + \mathbf{v}_t, \quad (1)$$

where $g(\cdot)$ is a (possibly non-linear) mapping describing the evolution of the state from one time step to the next, and \mathbf{v}_t is a random vector of process noise. This model is very general and can be used to capture a wide range of noise source dynamics. This sort of first-order model (where \mathbf{x}_{t+1} only depends on \mathbf{x}_t) is standard in the tracking and filtering literature. Examples of specific dynamic models are discussed in further detail in Section 4 below.

Measurements are obtained by a network of sensors. We assume there are n sensors, and the measurement obtained at sensor j at time t is given by

$$\mathbf{z}_t^{(j)} = h_{j,t}(\mathbf{x}_t) + \mathbf{w}_{j,t} , \quad (2)$$

where $h_{j,t}(\cdot)$ is the function describing the mapping from the current noise source state to the observation at node j at time t , and $\mathbf{w}_{j,t}$ is additive measurement noise at node j at time t . Again, the measurement model is general. Of particular interest in this project is the case where $h_{j,t}(\mathbf{x}_t)$ is the true bearing angle from the sensors position at time t to the noise source position.

3 Methodology

In our proposed approach, each node runs its own local instance of the tracking algorithm. Rather than having each node operate in isolation, with node j updating its state estimate using only its own measurements, our aim is develop schemes under which sensors share information about their local measurements so that the tracking performance of the overall system is superior to that of any individual sensor. The main issues to be addressed are then:

- What information should be communicated?
- What tracking algorithm should be used at each node?

Below we describe our approach to these issues. With each node running a local tracking algorithm, communication plays the role of synchronizing the state estimates across all nodes in the network so that, ideally, all nodes have the same estimate that would be computed by a single tracking algorithm that had direct access to all of the measurements. We use gossip algorithms (discussed in Section 3.1) to diffuse information across the network and drive the state estimates at each node to a consensus.

For the local tracking algorithm used at each node, we investigate two options: the particle filter and the ensemble Kalman filter (discussed in Sections 3.2 and 3.3). Particle filters are attractive when the dynamic model $g(\cdot)$ is non-linear (e.g., if the noise source may be highly manoeuvring), and when the measurement model $h_{j,t}(\cdot)$ is non-linear, as is certainly the case for the bearings measurement model considered in this project. However, the overhead associated with communicating the particle representation is non-trivial. Variations of the ensemble Kalman filter make it possible to reduce the communication overhead, possibly at the cost of some performance degradation.

3.1 Gossip algorithms for distributed averaging

We begin by discussing gossip algorithms, which we use to synchronize and spread information across the network. Gossip algorithms are a class of distributed message-passing algorithms for aggregating data over a network [1, 2, 3]; see [4] for a survey. We focus on gossip algorithms which solve the *distributed averaging problem*: each node $j = 1, \dots, n$ initially has a value $y_j(0)$, and the goal of the network is for all nodes to obtain the average of the initial values, $\bar{y} = \frac{1}{n} \sum_{j=1}^n y_j(0)$. In order to reach this goal, the nodes repeatedly transmit messages and modify their local estimate of the average based on messages they receive. This process can occur synchronously (all nodes transmit, and then they update simultaneously before the next round of communication) or asynchronously (where nodes update whenever they receive a message and then broadcast their new result periodically).

In the most general case, after k updates have occurred across the network, each node has a current estimate $y_i(k)$ for the average value. At the next update/iteration, suppose that node i receives messages from a subset of nodes $N_i(k) \subseteq \{1, 2, \dots, n\}$. For each node $j \in N_i(k)$, the message received by node i from j contains the value $y_j(k)$. Then node i updates its new value to

$$y_i(k+1) = A_{i,i}(k)y_i(k) + \sum_{j \in N_i(k)} A_{i,j}(k)y_j(k), \quad (3)$$

where the weights $\{A_{i,j}(k)\}_{j=1}^n$ typically satisfy (i) $A_{i,j}(k) \geq 0$; (ii) $A_{i,i}(k) > 0$; and (iii) $\sum_{j=1}^n A_{i,j}(k) = 1$. These conditions effectively state that (a) nodes form their new estimate by taking a weighted average of their old estimate with the estimates received from their neighbours, (b) they always give some weight to their own estimate, and (c) the total weight given to the estimates forming $y_i(k+1)$ is equal to 1.

Of particular interest in this project is the case where each node is only ever able to communicate directly with a subset of nodes, and this connectivity is represented as a graph $G = (V, E)$ with vertices $V = \{1, \dots, n\}$ corresponding to the nodes, and with an edge $(i, j) \in E$ if node i can receive a message from node j . In order for there to be any hope of the entire network converging to a consensus on the average, the graph of possible connections, G , should be *strongly connected*: there should be a path (possibly involving multiple hops) from every node i to all other nodes $j \in V \setminus \{i\}$ in the network.

For most gossip algorithms it can be guaranteed that all nodes converge, asymptotically, to a consensus on the average, and the rate of convergence depends on structural properties of G . More specifically, after k iterations we have (with high probability, in the case of asynchronous randomized gossip),

$$\sum_{i=1}^n (y_i(k) - \bar{y})^2 \leq \rho^k \sum_{i=1}^n (y_i(0) - \bar{y})^2, \quad (4)$$

where $0 < \rho < 1$ is a constant related to the connectivity of the graph G . Roughly speaking, if G is very well connected (e.g., there are many short paths between every pair of nodes),

then ρ is closer to 0 and the algorithm converges faster. Similarly, if there is a probability that packets are not successfully received then this increases ρ (bringing it closer to 1, corresponding to slower convergence).

Below we will discuss how gossip algorithms can be used as a subroutine within distributed tracking algorithms to synchronize the state estimates at each node. For distributed tracking, each node has its own local representation of the state estimate, and we would like to keep these representations synchronized. Nodes will update these local representations using their local measurement, and then they will perform some gossip iterations in order to re-synchronize (equivalently, to spread the information each node learned from its local measurement across the network). If the state estimates are well-synchronized to begin, and if the update using local measurement does not change the state estimate too much, then after a few gossip iterations they will be synchronized again.

3.2 Distributed particle filtering

The dynamic model (1) defines the distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ of the state at time t given the state at time $t-1$. Similarly, the measurement model (2) defines the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ of an observation \mathbf{z}_t given that the state is \mathbf{x}_t . Given these models, the Bayes-optimal recursion for the posterior state estimate is given by the equations:

$$\text{Predict:} \quad p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1} \quad (5)$$

$$\text{Update:} \quad p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}, \quad (6)$$

where $\mathbf{z}_{1:t-1}$ denotes the collection of all measurements from times 1 up to $t-1$. Except for under very special circumstances (e.g., unless the models $g(\cdot)$ and $h(\cdot)$ are both linear and the noise terms \mathbf{v}_t and \mathbf{w}_t are Gaussian), these recursions cannot be computed analytically. Hence, in most cases, we must resort to some approximation.

Particle filters approximate the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ using a set of m weighted particles, $\{\hat{\mathbf{x}}_t^{(i)}, w_t^{(i)}\}_{i=1}^m$; specifically,

$$\hat{p}_m(\mathbf{y}_t|\mathbf{z}_{1:t}) = \sum_{i=1}^m w_t^{(i)} \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t^{(i)}), \quad (7)$$

and where $\delta(\cdot)$ is the Dirac delta function. Each particle $\hat{\mathbf{x}}_t^{(i)}$ can be viewed as a candidate hypothesis for the current state value, and the corresponding importance weight $w_t^{(i)}$ is an approximation to the posterior density at $\mathbf{x}_t^{(i)}$. The particles are updated from one time step to the next (i.e., from $\hat{\mathbf{x}}_t^{(i)}$ to $\hat{\mathbf{x}}_{t+1}^{(i)}$) by simulating the noise source dynamic model (1), and then the new importance weights $w_{t+1}^{(i)}$ are updated based on the new position $\hat{\mathbf{x}}_{t+1}^{(i)}$ and the new measurements \mathbf{z}_{t+1} . For more background on particle filters, see [5, 6].

For distributed particle filtering, we consider the case where each node maintains its own set of weighted particles. Let $\mathbf{z}_t^V = \{\mathbf{z}_t^{(j)} : j = 1, \dots, n\}$ denote the collection of measurements from all nodes at time t . We assume that measurements at different nodes are conditionally independent given the state. The joint likelihood then factorizes into

$$p(\mathbf{z}_t^V | \mathbf{x}) = \prod_{j=1}^n p(\mathbf{z}_t^{(j)} | \mathbf{x}_t) .$$

Then, to incorporate information from all of the measurements, the weight of the i th particle should be set to

$$w_t^{(i)} \propto \exp \left(\sum_{j=1}^n \log p(\mathbf{z}_t^{(j)} | \hat{\mathbf{x}}_t^{(i)}) \right) , \quad (8)$$

where the proportionality constant only depends on the particle $\hat{\mathbf{x}}_t^{(i)}$, and not on the measurements. Hence, in the distributed setting, to fuse measurements gathered at all sensors we need to compute the sum in (8). Of course, after rescaling each term in the sum by n , this is equivalent to computing an average,

$$\frac{1}{n} \sum_{j=1}^n n \log p(\mathbf{z}_t^{(j)} | \hat{\mathbf{x}}_t^{(i)}) ,$$

and so we can use the gossip algorithms described in Section 3.1 with the value at node j initialized to $y_j^{(0)} = n \log p(\mathbf{z}_t^{(j)} | \hat{\mathbf{x}}_t^{(i)})$.

The performance of particle filters, in general, is related to the number of particles used. As the number of particles m tends to infinity, the particle approximation (7) converges to the posterior. On the other hand, in the setting of distributed particle filtering, we need to reach a consensus on each particle weight. Thus, each gossip message is of dimension m . Consequently, we have a tradeoff in distributed particle filtering, where increasing the number of particles m should improve the tracking accuracy at the cost of increased communications. Increasing m also increases the amount of computation each node must perform in order to update the particle locations and weights, but we believe this is a secondary concern in the context of this project, since the underwater communication medium is the main bottleneck limiting the number of particles that may be practical.

3.3 Distributed ensemble Kalman filtering

The distributed ensemble Kalman filter is an alternative approach to tracking, which we consider promising because it allows us to reduce the communication overhead [7]. The (centralized) ensemble Kalman filter [8] uses a set of samples, which we denote by $\mathbf{x}_t^{(i)}$, to approximate the first two moments of the state density. Two particular versions of the ensemble Kalman filter—the ensemble square root filter [9] and the deterministic ensemble

Kalman filter [9]—are particularly attractive for use in the distributed setting because their updates dramatically reduce the amount of communication. In these filters each of the samples is updated using a Kalman filter-like recursion,

$$\text{Forecast:} \quad \mathbf{x}_t^{(i)f} = g(\mathbf{x}_{t-1}^{(i)}) \quad (9)$$

$$\text{Update:} \quad \mathbf{x}_t^{(i)u} = \widehat{\mathbf{x}}_t^f + \mathbf{K}(\mathbf{y} - \mathbf{H}_t \widehat{\mathbf{x}}_t^f) + \mathbf{T}(\mathbf{x}_t^{(i)f} - \widehat{\mathbf{x}}_t^f), \quad (10)$$

where $\widehat{\mathbf{x}}_t^f = \frac{1}{m} \sum_{i=1}^m g(\mathbf{x}_{t-1}^{(i)})$ is the average of the forecasted samples (i.e., the average state after passing each sample through the dynamic model), \mathbf{H}_t is the linear measurement matrix. $\mathbf{K} = \mathbf{P}_t^f \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t^f \mathbf{H}_t^T + \mathbf{R})^{-1}$ represents the Kalman gain, in which \mathbf{R} is the measurement noise covariance matrix and $\mathbf{P}_t^f = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_t^{(i)f} - \widehat{\mathbf{x}}_t^f)(\mathbf{x}_t^{(i)f} - \widehat{\mathbf{x}}_t^f)^T$ is the prediction covariance matrix, estimated from the samples. The operator $\mathbf{T} = (\mathbf{I} - \mathbf{K}\mathbf{H}_t)^{1/2}$ for the ensemble square root filter, where \mathbf{I} is the identity matrix, and $\mathbf{T} = \mathbf{I} - \frac{1}{2}\mathbf{K}\mathbf{H}_t$ for the deterministic ensemble Kalman filter. These equations are suitable for linear measurement models, but for non-linear measurement models, we perform a linearization of each sensor's measurement function $h_{j,t}(x^{(i)})$ to derive a matrix $\mathbf{H}_t^{(i)}$ (due to the linearization, this matrix can vary for each sample i).

The main challenge in formulating a distributed update step lies in the fact that the forecast and update equations (9)-(10) require the knowledge of the complete set of measurements, measurement functions and measurement noise statistics. Sensor nodes only have access to their local measurements and may only be aware of their own measurement model parameters. In [7], we addressed this challenge by expressing the update equation in an alternative information filter form. We identify summary statistics that can be computed by global network averaging of locally-computable information using gossip procedures.

4 Results

4.1 Distributed particle filtering

Next we summarize the results obtained for distributed particle filtering. A preliminary version of our results are presented in the conference paper [10], and an extended version is in preparation to be submitted to a journal shortly.

To evaluate the performance of the proposed gossip-based distributed particle filtering solution, we use data from an at-sea experiment conducted in October 2012 in the Emerald Basin area of the Scotian Shelf [11]. The experimental setup involves five acoustic sensors, a single acoustic source, and an echo-repeater in the water. We focus on localizing and tracking the source using measurements obtained at the sensors. Figure 1 depicts the trajectory of the source along with the trajectories of the sensors, which were freely drifting. The experiment lasts for roughly 180 minutes, with each sensor providing measurements once per minute. During this time, the noise source makes two cycles.

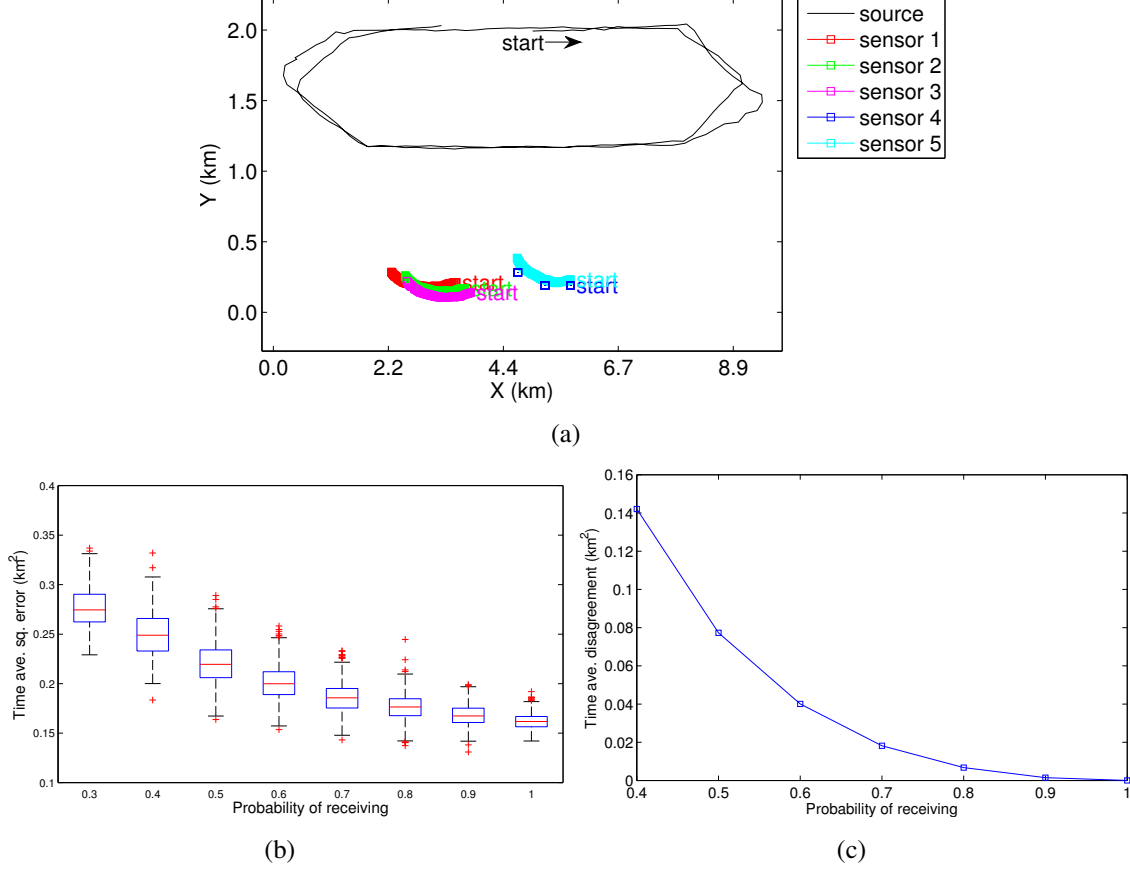


Figure 1: (a) Trajectories of the source and the sensors in the experiment considered. (b) Time-averaged squared error vs. reception probability p for a particle filter using both bearings and range measurements. (c) Time-averaged disagreement vs. reception probability p . The results shown correspond to 1000 Monte Carlo trials.

We consider two dynamic models. The first models the noise source motion at each time-step as either remaining true with a constant velocity (with probability p_{CV}) or making a turn with constant turning rate (with probability $1 - p_{CV}$). The approach combines a constant velocity model (where the source moves along a straight line) and a coordinated turn, so we refer to it as the CV-CT model. The second model we consider is a simple random walk model (which we refer to as the RW model), where the motion of the noise source at each time step is random and isotropic about its previous state.¹

Our preliminary investigation, reported in [10], considered the case where each sensor obtains noisy bearings measurements as well as a noisy range measurement. For this setting,

¹In both dynamic models, we model the process noise \mathbf{v}_t and measurement noise \mathbf{w}_t as being zero-mean Gaussian, with a diagonal covariance matrix whose parameters were estimated from the data [10]. In the CV-CT model, we obtain estimates for the turn probability and the turn radius from the dataset.

we simulate the case where each time a sensor transmits a message it is received at each other sensor independently with some probability p , and we study the tracking performance as a function of the reception probability. This is shown in Figure 1(b), for a filter with 1600 particles, and Figure 1(c) shows the time-averaged disagreement² between nodes as a function of p , under the CV-CT model. Figure 1 shows that the message reception probability has a non-trivial impact on the performance. The time-average squared error clearly increases when the reception probability is lower. As the probability of receiving messages from other sensors approaches zero, the performance corresponds to that of each sensor performing tracking in isolation. Clearly collaboration is beneficial, both for reducing the mean time-averaged square error as well as reducing the error variance. From Figure 1(c), we also see that as there is less collaboration (i.e., when sensors receive fewer messages from their neighbours) there is greater disagreement among the state estimates at different sensors.

Next we focus on the performance when sensors only use bearings measurements. Figure 2(a) shows the *root mean-squared error* (RMSE) as a function of the number of particles used in a centralized particle filter (i.e., one having access to the measurements from all sensors), and Figure 2(b) shows the computation time per filter update as a function of the number of particles. Note that the horizontal axis of these figures is linear over the range from 10 to 100, and then it increases geometrically from 100 to 1600. The experiment was conducted on a computer with a 1.7GHz Intel i7 processor with 8GB of RAM. The performance of the centralized filter gives a benchmark for what we can hope to achieve in the distributed setting. From Figure 2(b) we observe that the computation time scales linearly with the number of particles used. Figure 2(c) shows the RMSE over time of the centralized particle filter as a function of time for the two different dynamic models. The average error is lower for the CV+CT model, which makes stronger assumptions about the noise source dynamics, but the CV+CT model does have higher peak error.

To better understand the cause of these peak errors under the CV+CT model, we observe the timing of these errors coincides with times when the noise source is making a turn towards the sensors, as shown in Figure 3. During these time periods, the noise source trajectory is roughly co-linear with the sensor positions. Bearings-only tracking is known to be especially challenging when the source moves in a line which is co-linear with the sensors, since then the bearings-measurements do not change significantly although the source may have moved a large distance. This suggests that it is also important to consider the problem of sensor placement.

²Here the disagreement is the average, over all pair of nodes, of the square difference between the nodes' maximum a posteriori state estimate.

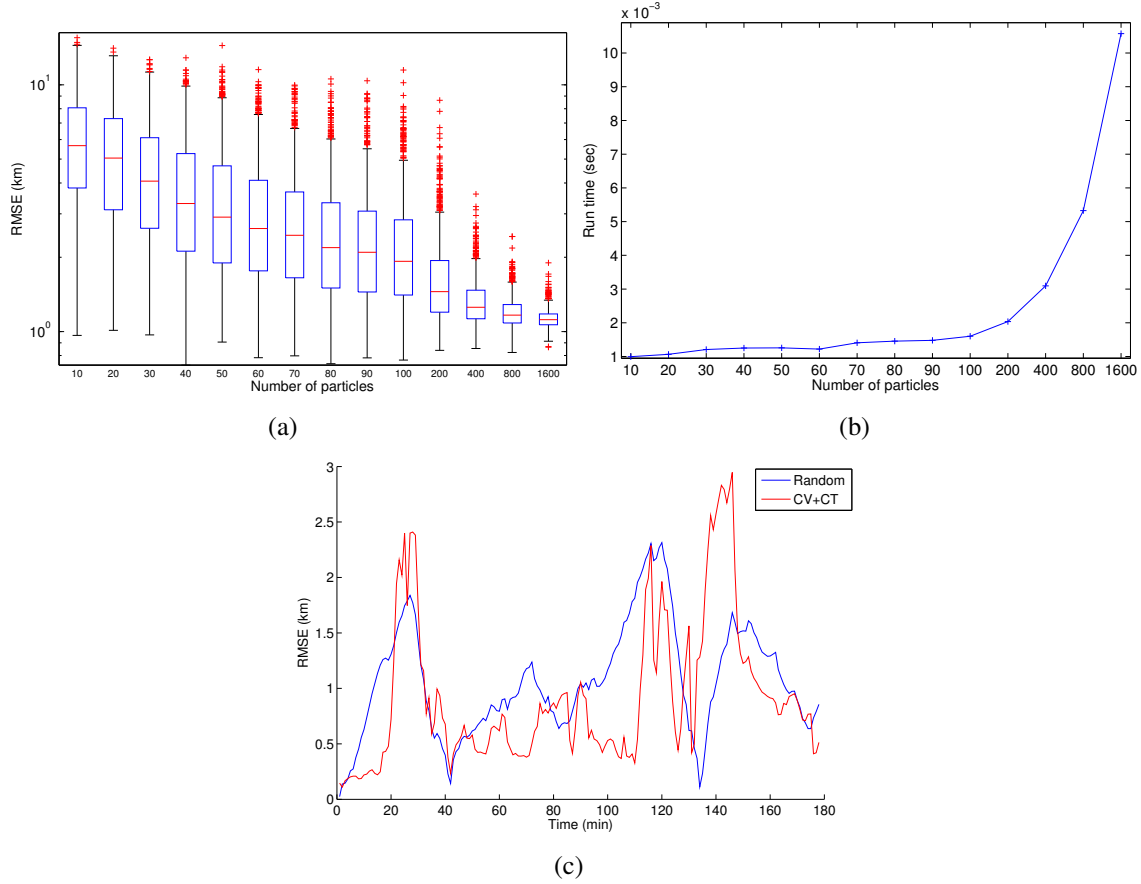


Figure 2: Performance of the centralized particle filter with the number of particles ranging from 10 to 1600. (a) Root mean-squared error vs. number of particles. (b) Runtime vs. number of particles. (c) Example tracking performance under the two different motion models for the centralized particle filter with 1600 particles.

4.2 Distributed ensemble Kalman filtering

In order to examine the performance of the proposed distributed ensemble Kalman filter, we simulate a nonlinear *radio-frequency (RF) tomography* tracking system (see [12] for a complete description of the RF tomography model). Twenty four nodes are deployed at the boundary of a square region of $50\text{m} \times 50\text{m}$. The sensor nodes track a single target for 50 seconds, making a set of measurements every second. Sensor nodes can communicate with each other if their separation is less than a transmission range of 15m. The target motion is modeled by the nearly coordinated turn model, which assumes unknown cartesian velocity but known turn rate.³ Table 1 compares the tracking accuracy of the distributed ensemble Kalman filters we provided and state-of-the-art non-linear distributed tracking alternatives.

³We use the RF tomography measurement model from [12], with parameters $\sigma_\lambda = 0.05$ and $\phi = 5$, and measurement noise standard deviation of 0.50. The standard deviation of the target motion is 0.25.

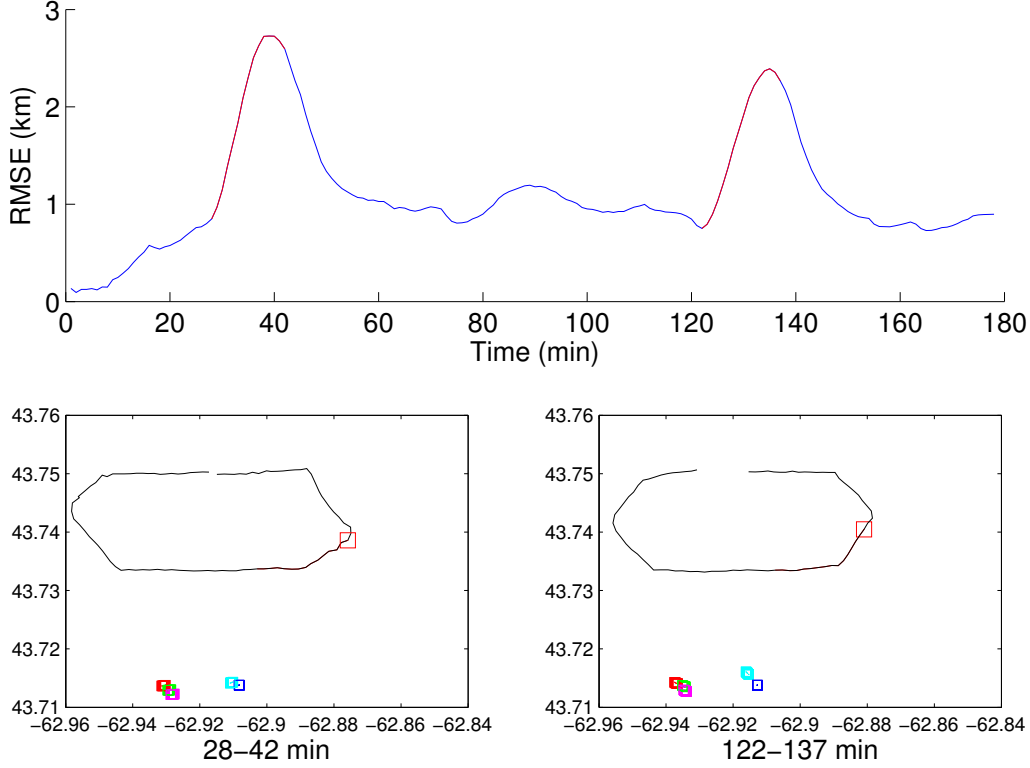


Figure 3: RMSE of the particle filter under the CV+CT model. Red segments indicate times where the RMSE peaks. These coincide with times when the noise source turns towards the sensors and its trajectory is roughly co-linear with the sensors.

As we reduce the number of scalars that each node is permitted to transmit, we see that the proposed algorithms clearly outperform the alternative methods.

5 Summary and ongoing work

We have investigated the problem of distributed target tracking. Our main application of interest involves bearings-only measurements using a network of multiple underwater acoustic sensors. The challenges of underwater communication have motivated us to focus on distributed tracking algorithms that are capable of addressing the non-linearities in both target dynamics and sensor measurements. We have therefore concentrated our attention on distributed particle filtering and distributed ensemble Kalman filtering.

We have conducted a thorough performance assessment of candidate distributed particle filtering algorithms using experimental datasets. To the best of our knowledge, this is the first time that decentralized particle filters have been carefully compared using experimental data. Our research has highlighted the need for improving the distribution of the sensors

Table 1: Tracking Performance for the Distributed Ensemble Kalman Filter

| | Ave. RMSE \pm Std. Dev. (% track loss) | | |
|-----------------------------|--|-----------------------------------|--------------------------------------|
| Scalars | 1000 | 500 | 200 |
| Gaussian Approximation [13] | 0.17 ± 0.01 | $0.20 \pm 0.02(2)$ | $0.35 \pm 0.18(22)$ |
| Likelihood Consensus [14] | 0.33 ± 0.22 | - | - |
| Set Membership [15] | $0.26 \pm 0.18(3)$ | $0.33 \pm 0.19(7)$ | $0.36 \pm 0.25(16)$ |
| Top-m Selective Gossip [16] | $0.27 \pm 0.22(2)$ | $0.34 \pm 0.20(11)$ | $0.70 \pm 0.41(50)$ |
| Distributed UKF [17] | $0.21 \pm 0.09(18)$ | $0.61 \pm 0.49(74)$ | $0.40 \pm 0.26(95)$ |
| Distributed ESRF | 0.19 ± 0.05 | 0.18 ± 0.05 | $0.22 \pm 0.10(6)$ |
| Distributed DEnKF | 0.23 ± 0.09 | 0.21 ± 0.08 | $0.22 \pm 0.09(1)$ |
| Centr. PF | 0.10 ± 0.01 | 0.10 ± 0.01 | 0.10 ± 0.01 |

to enhance observability of the targets, and the need to develop algorithms that can maintain a reasonable accuracy but have much less communication overhead. With this in mind, we have proposed a novel distributed filtering algorithm, developing techniques for using gossip algorithms to implement ensemble Kalman filtering calculations. Compared to other state-of-the-art non-linear tracking algorithms, simulation results suggest that the proposed filters can maintain reasonable accuracy while reducing the communication requirements.

In the next year of the project, we will migrate our attention to the multi-target tracking problem, which poses additional data association challenges and involves tracking in a much higher dimensional state-space. We are currently developing techniques for implementing distributed multi-sensor random finite-set filters.

References

- [1] Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006), Randomized gossip algorithms, *IEEE Trans. Information Theory*, 52, 2508–2530.
- [2] Aysal, T., Yildiz, M., Sarwate, A., and Scaglione, A. (2009), Broadcast gossip algorithms for consensus, *IEEE Trans. Signal Processing*, 57(7), 2748–2761.
- [3] Shaochuan, W. and Rabbat, M. (2013), Broadcast gossip algorithms for consensus on strongly connected digraphs, *IEEE Trans. Signal Processing*, 61(16), 3959–3971.
- [4] Dimakis, A. G., Kar, S., Moura, J. M., Rabbat, M. G., and Scaglione, A. (2010), Gossip algorithms for distributed signal processing, *Proceedings of the IEEE*, 98(11), 1847 – 1864.
- [5] Ristic, B., Arulampalam, S., and Gordon, N. (2004), Beyond the Kalman filter: particle filters for tracking applications, Artech House.
- [6] Doucet, A. and Johansen, A. M. (2009), A tutorial on particle filtering and smoothing: Fifteen years later, In *Handbook of Nonlinear Filtering*, Ch. 12, pp. 656–704, Oxford, UK: Oxford University Press.
- [7] Shahid, A., Üstebay, D., and Coates, M. (2014), Distributed ensemble Kalman filtering, (Technical Report) McGill University, Montreal, Quebec.
- [8] Evensen, G. (1994), Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *J. Geophys. Res.*, 99(C5), 10 143–10 162.
- [9] Sakov, P. and Oke, P. (2008), A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters, *Tellus A*, 60(2), 361–371.
- [10] Yu, J.-Y., Üstebay, D., Blouin, S., Rabbat, M. G., and Coates, M. J. (2013), Distributed underwater acoustic source localization and tracking, In *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, USA.
- [11] (2014), The Scotian Shelf in context, In *State of the Scotian Shelf Report*, Department of Fisheries and Ocean, Canada.
- [12] Nannuru, S., Li, Y., Zeng, Y., Coates, M. J., and Yang, B. (2013), Radio frequency tomography for passive indoor multi-target tracking, *IEEE Trans. Mobile Computing*, 12(12), 2322–2333.
- [13] Oreshkin, B. and Coates, M. (2010), Asynchronous distributed particle filter via decentralized evaluation of Gaussian products, In *Proc. ISIF Int. Conf. Information Fusion*, Edinburgh, UK.

- [14] Hlinka, O., Slučiak, O., Hlawatsch, F., Djurić and, P., and Rupp, M. (2010), Likelihood consensus: Principles and application to distributed particle filtering, In *Asilomar Conf. Signals, Systems and Computers*, pp. 349 –353.
- [15] Farahmand, S., Roumeliotis, S., and Giannakis, G. (2011), Set-membership constrained particle filter: Distributed adaptation for sensor networks, *IEEE Trans. Signal Processing*, 59(9), 4122 –4138.
- [16] Üstebay, D., Coates, M. J., and Rabbat, M. G. (2011), Distributed auxiliary particle filters using selective gossip, In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Proc. (ICASSP)*, pp. 3296–3299, Prague, Czech Republic.
- [17] Li, W. and Jia, Y. (2012), Consensus-based distributed multiple model UKF for jump Markov nonlinear systems, *IEEE Trans. Autom. Control*, 57(1), 227–233.